

Class structure

March 8, 2020

Schedule

See syllabus and course website for a detailed schedule.

- ▶ Bruteforce (1 week)
- ▶ Meet in the Middle (2 weeks)
- ▶ Differential Cryptanalysis (3 weeks)
- ▶ Linear Cryptanalysis (3 weeks)
- ▶ High performance computing (1 week)

At the end we will have 1 or 2 weeks to reiterate on previous subjects or discuss new topics, see the website for the most actual schedule.

Contact

Send an email to:

elambooi@campus.haifa.ac.il ¹

Or ask your question before/during/after lecture.

¹Yes, this is my email address... I did not make a spelling mistake

Exercises

The course is hands on heavy, so you are expected to be able to program the attacks we discuss in class.

- ▶ Weekly exercises
- ▶ Work in pairs
- ▶ Do **not** get behind

We use a system for you to check if the solution you have is correct. You can find the system at:

`https://cryptanex.hideinplainsight.io`.

Submitting exercises

- ▶ For every exercise hand in the following:
 - ▶ Compilable source code that implements the attack.
 - ▶ A (short) report on the attack.
 - ▶ Your handle on the exercise site.
 - ▶ If you work in pairs add both student id's to the report.
- ▶ Zip all the files and name the zipped folder (individual):
[studentID]_exercise_[exercise number].zip.
- ▶ Zip all the files and name the zipped folder (team):
[studentID1]_[studentID2]_exercise_[exercise number].zip.
- ▶ Exercises need to be submitted **before** the start of the lecture.
- ▶ Only submit code/reports you have written yourself.

Writing code

- ▶ Try to keep everything simple, readable and well documented. Keep in mind that you are writing code for the reader and not the computer.
- ▶ For all exercises C should be good enough (no need for ASM).
- ▶ Please use git or svn and back up often.
- ▶ Please keep extra requirements to a minimum (such as libs, compilation flags, etc.).
- ▶ If you need help, ask early.

Writing the report

The report should contain the following things:

- ▶ Your student ID (and your partner's if you do the exercise in pairs).
- ▶ A brief explanation of the attack you implemented.
- ▶ A brief explanation on the optimizations/datastructures implemented.
- ▶ How to build the program (a makefile makes everyones life easier).
- ▶ Any problems encountered during the implementation of the attack.
- ▶ (Extra) provide comparisons between different approaches, figures showing the asymptotical behaviour, extensions to more difficult problems, etc.

Grading

For the weekly exercises we employ the following grading scheme:

- ▶ If you solved the exercise you start with a 70 otherwise a 40
- ▶ Nice, clear, concise report (max +/-10)
- ▶ Nice, clear, readable code (max +/-20)
- ▶ Improved attack, advanced use of data structures, interesting approach (max +20)
- ▶ The maximum points you get for an exercise is 100

Final Course Grade

The final grade for the course is computed as follows:

HOME = average of 7 best homework assignments.

FINAL = grade for the final exercise.

$$\text{Course grade} = \frac{\text{HOME} + \text{FINAL}}{2}$$

Final project

More information will be given soon.